Seminar - Self-tuning Databases

Runtime Statistics
Self-tuning Histograms

Rainer Habrecht

habrecht@cs.uni-magdeburg.de

December 10, 2003

Overview

Introduction (1) - ST-Histograms

- Database systems require knowledge of the datas distribution
- Histograms are used in most commercial database systems
- High costs of building, maintaining or rebuilding
- One-dimensional and multi-dimensional histograms exist, but often only one-dimensionals are used (attribut value independence assumption)
- Difficult to choose the right subset of histograms for attributes or attribute combinations
- Static histograms do not recognize changes of the data distribution
- [CR94] introduced the usage of the query execution engines feedback

Introduction (2) - Approaches

- One-dimensional, static histograms
  - Equi-width Histogram
  - Equi-depth Histogram (depth - sum of frequencies)
  - MaxDiff(V,A)
- Multi-dimensional, static histograms
  - Extensions of one-dimensional algorithms
  - MHist [PI97]
  - GenHist
- ST-Histograms [AC99]
  - Main part here (later termed STGrid)
- STHoles [BCG01]
  - Only a short overview

STGrid (1) Introduction

- Done as part of AutoAdmin from Microsoft
- Low-cost information from query execution engine
- On-line or off-line operation (static histogram construction is always an off-line operation)
- Attribute dimensions
  - One-dimensional
  - Multi-dimensional
- Three phases
  - Constructing the initial histogram with simple methods
  - Refining bucket frequencies
  - Restructuring

STGrid (2) One-dimensional - Initial Histogram

- Number of buckets B, number of tuples T and a value range $[min, max]$ of attribute a is needed
- Buckets are evenly spaced between min and max
- Frequencies are set to T/B (uniformity assumption)
- min and max could be an approximation
- Other additional information (domain constraints, min-max-value from query workload) could be used

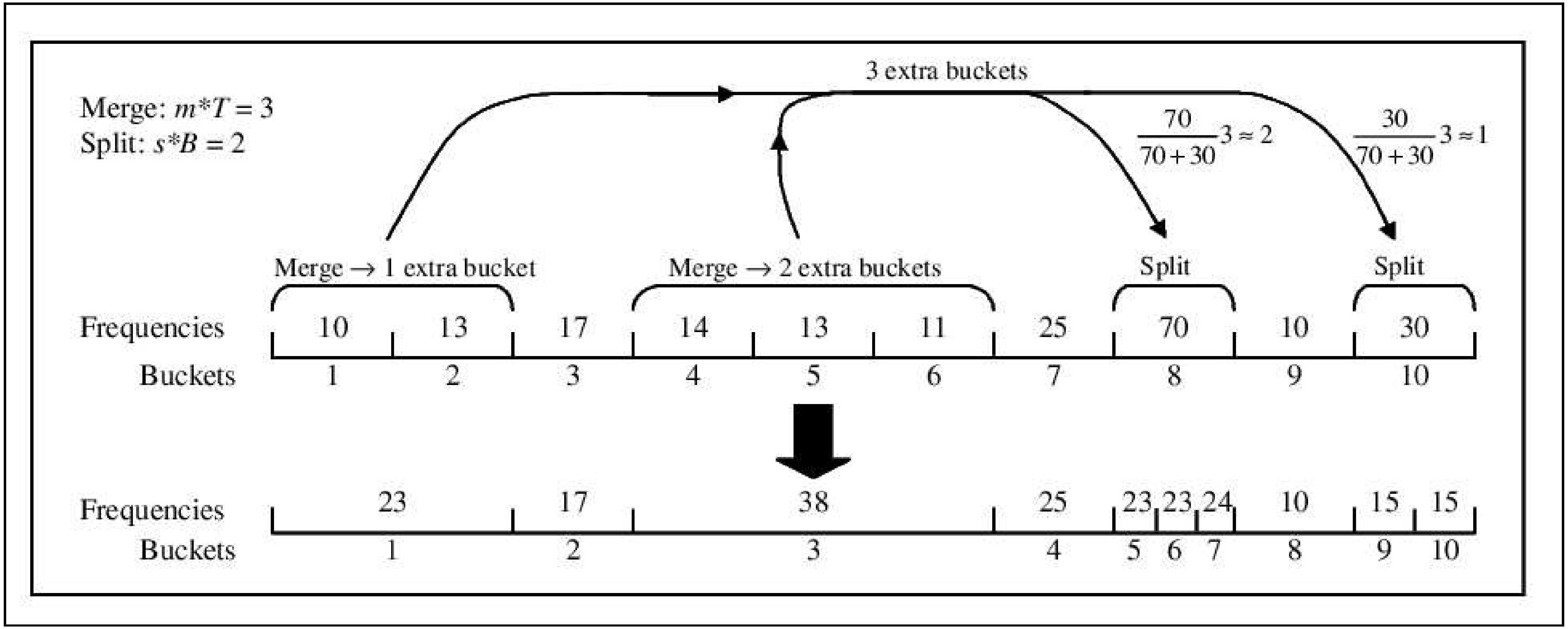
STGrid (3) One-dimensional - Refining Bucket Frequencies

- Use feedback information from queries of the workload
- Absolute estimation error is the difference between actual and the estimated result size ($esterr = act - est$)
- $esterr$ allows to distinguish between over- and underestimation
- Difficult to decide how to distribute the "blame" for the error; here proportional to the current frequency of the buckets
- $frac(b_i) = \frac{min(rangehigh, high(b_i)) - max(rangelow, low(b_i)) + 1}{high(b_i) - low(b_i) + 1}$
- $freq(b_i) = max(\frac{freq(b_i) * (1 + \alpha * esterr * frac(b_i))}{est}, 0)$

STGrid (4) One-dimensional - Restructuring

- Moving bucket boundaries, avoids grouping of high frequency and low frequency values in the same bucket
- Number of freed buckets by merging is the same as the number of created buckets by splitting
- Merging
  - Consecutive buckets with low frequency ($m \leq 1\%$)
  - Greedy algorithm, starting with B single buckets, finding runs of low frequency buckets
- Splitting
  - $s\%$ of the high frequency buckets are split (s split threshold)

STGrid (5) One-dimensional - Restructuring



STGrid (6) Multi-dimensional - Initialization, Refinement

- Initial Histogram
  - Extension of one-dimensional initialization also with uniformity and independence assumption for all dimensions
  - Grid structure, termed frequency matrix
  - Alternative: use existing one-dimensional histograms as starting point
- Refinement
  - Identical with one-dimensional case
  - But the overlapping fraction is now a volume of the region

STGrid (7) Multi-dimensional - Restructuring

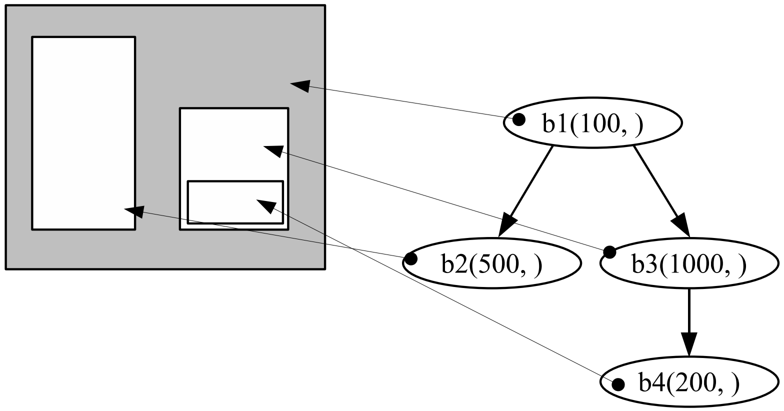
- Same parameters m and s
- Also based on merging buckets with similar frequencies and splitting high frequency buckets
- Modification of the dimensions are independent to each other
- Process one dimension at a time with the above one-dimensional algorithm
- Adapted algorithm to satisfy the requirements of the n-dimensionality

STGrid (8) Experiments and Conclusions

- Zipfian distributed data set used (parameter z describes the skew/correlation of data)
- One-dimensional accuracy is superior to uniformity assumption and inferior to static histograms (here MaxDiff(V,A))
- Multi-dimensional accuracy
  - Superior to uniformity assumption
  - With low correlation ($z \leq 1$) STGrid is more accurate than MHist
  - But $z > 1$ leads to inaccuracy (Relative error $> 20\%$ for two dimensions)
- Convergence is given for both on-line and off-line refinement and is fairly rapidly

STHoles

- New partitioning scheme with overlapping and nested buckets
- Grid constraint of STGrid is too rigid
- Buckets could be nested into another buckets
- Modeling complex shapes of buckets not restricted to rectangles
- Tree structure with buckets as nodes
- Refinement and restructuring are more complex, but similar to STGrid



Conclusion

- Self-tuning histograms are good for low to moderate correlated data distribution
- STGrid is the first multi-dimensional self-tuning approach
- STHoles creates the better shape of the data distributions model
- Multi-dimensional self-tuning histograms should be the first choice
  - On-line refinement
  - Fairly rapid convergence
  - Acceptable overhead (approx. 10%)

References

[AC99] Ashraf Aboulnaga and Surajit Chaudhuri. Self-tuning histograms: building histograms without looking at data. In *Proceedings of the 1999 ACM SIGMOD*, pages 181–192, 1999.

[BCG01] Nicolas Bruno, Surajit Chaudhuri, and Luis Gravano. STHoles: a multidimensional workload-aware histogram. In *Proceedings of the 2001 ACM SIGMOD*, pages 211–222, 2001.

[CR94] Chung-Min Chen and Nick Roussopoulos. Adaptive selectivity estimation using query feedback. In *Proceedings of the 1994 ACM SIGMOD*, pages 161–172, 1994.

[MD88] M. Muralikrishna and David J. DeWitt. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries. In *Proceedings of the 1988 ACM SIGMOD*, pages 28–36, 1988.

[PI97] Viswanath Poosala and Yannis E. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *VLDB'97, Proceedings of 23rd VLDB*, pages 486–495, 1997.

Questions ?